Discretionary Access Control

This module describes the concept of discretionary access control(DAC), describes the TCSEC requirements for DAC, and then goes into some DAC mechanisms that can be used to implement a DAC policy in a trusted system.

Module Learning Objectives

The material presented in this module can be read independently of theother modules. Upon completion of this module, the student should:

1. Understand DAC.
2. Understand the TCSEC DAC requirements.
3. Be familiar with some implementations of DAC.
4. Be familiar with certain issues of the implementation that require special attention.

Overview

DAC is a security mechanism that allows a user to specify explicitly the types of access that other users may have to information under that user's control. For example, a DAC mechanism allows the owner of an object (e.g., file, directory, etc.) to specify each user that can or cannot read or write the object. DAC is discretionary because control of access to the object is at the`discretion' of the user. DAC is required at all classes in the TCSEC. The TCSEC definition of DAC describes how users can be restricted to specific access modesfor data, but does not include the notion of restricting specific applications to specific access modes for data.

The Bell-LaPadula security policy model [Bell76] describes discretionary access permissions as a matrix, with all the subjects on one axis and allthe objects on the other axis. Each cell of the matrix is used to describe the access permissions of the corresponding subject to the corresponding object. For example, a particular cell could indicate that read and execute permission is allowed, but write permission is not. The TCSEC demands this form of DACat B3 and A1. At B2 and below, a less flexible model, often based on the UNIX user/group/world design, is acceptable.

The possible DAC rights that can be controlled, sometimes referred to as “modes of access,” are considerable. The TCSEC does not specifically mandate that a certain set be required. The three most common modes of accessare read, write, and execute. Additional rights could control operations like append, delete, move, etc. [DAC87] provides greater detail in this area.

The Named Object Issue

Although it would seem to be a fairly straightforward concept, the identification of named objects has caused some debate amongevaluators . We will not solve this issue here, just describe it. The TCSEC states that “the TCB shall define and control access between named users and named objects(e .g., files and programs) in the ADP system.” Although two examples are provided, the term “named object” is not defined. Whether or not inter-process

communications constructs (e.g., pipes in UNIX) are named objects has been the basis of considerable discussion. Related is the question of whatlevel of granularity is appropriate for DAC. If a user can access a specific set of bytes in virtual memory by referring to them by number, does that constitute a name, or does the word “name” in the term “named object” imply ahigher level of abstraction?

DAC Limitations

There are some inherent problems with DAC. A DAC mechanism can accidentally or deliberately be misused or abused. Users can provide toomuc h access to other users since access is under the user's control. Inaddition, users with access to an object can effectively pass their rights on to otherusers by either performing the operation on behalf of the other user, or making a copy of the object and allowing other users to perform the operation on the copy. This is because the access permissions of the new object are typically notinherited from the original object.

Discretionary access rights can also be propagated unbeknownst oragainst the wishes of the owner. If a user grants read access to an object, then those users that have read access can make a copy of that object and then grant otherusers any type of access to the copy of the object. This vulnerabilityrequires users to trust other users to not reveal information that the owner would not wishto have revealed. If a user grants write access to an object, then thoseusers cannot only change the object for their own reasons, but they could also change it on behalf of another user. However, they could not give another user direct modify access to the object.

In some ways, certain access rights cannot be completely revoked once granted. The most notable of these is read access. Once read access is granted, users can make copies of the object. Unless the system tracks where copies of data are stored (which is unlikely and difficult), revoking read access to the objectwill make the object and any updates to it unreadable, but will not affect the access rights to any existing copies of the object.

The most insidious form of access rights propagation is not what can bedone by other users but what can be done by software. When a user executes a piece of software, that software, during its execution, acts on the user's behalf and assumes all rights and privileges of the user while doing so. Most software has been written to do what the user wants, but the growing proliferation of computer viruses (e.g., virus, worm, Trojan horse) has made this a significant threat. These programs rely on the fact that all a user's access rightsare granted to the software the user executes. These ill-gotten rights are then used to access programs (to propagate itself) and data (for whateverreasons) in a manner that the user would normally not want.

TCSEC Requirements

The TCSEC requires that all trusted systems, from C1 on up, enforce DAC. At C1, it requires that the TCB control access between named users andnamed objects. This sharing can be controlled to the level of named users and/or named groups. At C2, the DAC requirement evolves to further specify that it

be controlled to the level of named individuals and/or named groups of individuals as well as requiring that the TCB provide controls to limitthe propagation of access rights. “Limiting the propagation of access rights” means that if user “A” grants user “B” write access to a file, user “B” does not gain the ability to grant access to a third user.

There is a requirement to limit the propagation of access rights at C2and above. That requirement explicitly ensures that the mechanism used to support the DAC enforcement is wholly under the control of the TCB. In other words, access to an object cannot somehow be granted outside of the TCB. For example, the use of a passwords as the sole file access control mechanism, would not meet the requirement to limit the propagation of access rights. The passwords required for access control could be disseminated on paper orby word of mouth (hence outside the control of the TCB), and thereforesubject to disclosure and further unknown abuse.

At B3, the DAC requirements are enhanced. The TCSEC specifies that theTCB shall be able to grant or withhold discretionary access to an arbitrarysubset of users. The user/group/world/model, usually implemented as permission bits, does not allow for this.

The B3 requirement is often implemented by an access control list(ACL). One will often hear it said that “ACLs are required at B3,” although the TCSEC does not demand a specific implementation, but requires a mechanism that shall be able to specify a “list of individuals and groups of namedindividuals with their respective modes of access.” In addition, the mechanism shall be able to specify a “list of named individuals, and a list of groups of named individuals for which no access to the object is given.”

DAC Implementations

Discretionary access rights are typically granted by explicit accessmodes suc h as read, write, execute, delete, append, etc. These rights can be granted through a variety of mechanisms such as permission bits, access control lists, capabilities (these are described later). These rights areexplicitly granted by the controller (typically the owner) of an object to other users. The “owner” is initially the user who created the object. Some systems also provide theability for a user to share or transfer ownership or control of an object.Typically , shared control can be revoked. But, once sole ownership or control isgiven away, control can only be returned by the user to which the rights were given. A user with shared control usually can grant and revoke access rightsjust like the original owner. Depending on the implementation, a user with shared control may or may not be able to affect the rights granted or denied by the original owner. If not, then the other user is considered subordinate to the original owner, not co-equal.

DAC mechanisms typically allow an object's owner to specify the DAC permissions for the object, although this is not required. Interpretation[I-0020] states:

> “A TCB need not provide all users with the capability to control the sharing of objects. A DAC policy where only system administrators assign access to objects can satisfy the DAC requirement. The SFUG

> shall clearly identify the roles or user types (e.g., systemadministrator) who can control sharing."

In other words, systems can allow certain privileged users to change an object's DAC settings, either exclusively or in addition to granting that capability to ordinary users.

The varieties of DAC mechanisms are quite numerous. UNIX uses permission bits to control discretionary access to objects. For each object there are three groups of three bits. Each group controls read, write, and execute access to the object. The three groups represent the object's owner, the object's group, and all others. (An object's group is the object's owner's group). If the owner'sexecute bit is set, then the DAC mechanism will allow the owner to execute that object (presumably a program). If the other's read bit is set, then allother users can read the object, etc. The UNIX group mechanism allows subsets of the users on a system to be granted access.

Every user can be assigned to a group or set of groups, and permission to access a file can be granted or withheld to one's fellow group members. However, this mechanism does not allow a user to grant or withhold access permission toan arbitrary subset of a group, or some combination of two or more groups. This capability distinguishes B3 and A1 DAC from the DAC required at B2 and below. An ACL is a list associated with each object that describes the access rights to that object. The list describes which users, or groups, can read, write, execute, append, etc. the object. The dual of ACLs are capabilities. Instead of associating the access rights with each object, capabilities associate access rights with the users that hold the rights. A user's capabilities describe the operations that the user can perform on specific objects.

ACLs and capability lists allow arbitrary sets of users to be granted ordenied access to an object. Both of these mechanisms have their own strengths and weaknesses. Both have to deal with the same issues, such as propagation of access rights and revocation. For an ACL system, if a user is deleted, the user's name needs to be removed from all the ACLs on the system. This removalcan be a considerable task, as there can be thousands or millions of objectson a system. For capabilities, a similar problem occurs when an object is deleted. The object name must be removed from all users' capabilities. This removal may be somewhat easier to perform since there are typically a reasonablysmall number of users. However, objects are removed from a system frequently so this removal would need to be performed quite often. For the ACL system, users are removed infrequently, but the cost is great for each removal. Additional information on ACLs and capabilities can be found in [Saltzer75].

Some DAC mechanisms can be used to create what are called "protected subsystems." A protected subsystem is a means of encapsulating applications and data in such a way that a group of procedures or programs may have exclusive access to a group of objects. In this manner, an application layer policy can be constructed to enforce an additional set of accessconstraints on the objects controlled by the programs. The programs can enforce nearly any access control policy desired. However, in practice, these policies should not violate the DAC or MAC policies, unless the application is trusted. An example of a protected subsystem is a database manager that controls allaccesses to the

data in a database. The database manager can control who can access or modify the database according to an internal algorithm that is independent ofthe DAC mechanism enforced by the operating system. Additional description can be found in [DAC87].

Delayed Revocation of Access Rights

The Bell-LaPadula model notes that "a state satisfies the [discretionary security] property provided every current access is permitted by thecurrent access permission matrix." An issue brought about by this property is that of revocation. What do you do when the access permission matrix is changed? A recent Interpretation [I-0002] specifies:

> A TCB is not required to provide any mechanism for the immediate revocation of DAC access to an object where access has already been established (e.g. opened) when access to that object is reduced. Itis sufficient for the SFUG and other documentation to describe the product's revocation policy. However, a change in DAC permissions shall have an immediate effect on attempts to establish new access to that object.

In other words, the DAC policy of a secure system may allow for a delay in the revocation of access rights if a subject is currently exercising theaccess right at the time that access is being withdrawn. The exact property enforced by a system should be clearly defined so that designers, developers, implementors and users all understand the exact discretionary security policyenforced by the system.

Relevant Trusted Product Evaluation Questionnaire Questions

2.1 SUBJECTS

A subject is an active entity in the system, generally in the form of aperson, process, or device that causes information to flow among objects or changes the system state. A subject can be viewed as a process/domain pair whose access controls are checked prior to granting the access to objects.

C1:

4. (a) What are the security attributes of a subject? (Examples of security attributes are user name, group id, sensitivity level etc.) For each type of subject in your system (i.e., user, process,device, etc.), what mechanisms are available to (b) define and (c)modify these attributes? (d) Who can invoke these mechanisms?

5. (a) What are other privileges a subject can have? (Examples of such privileges are: super user, system operator, system administrator, etc. Your operating system may assign numerous other privileges to the subjects, such as the ability to use certain devices.) For each type of subject in your system, what mechanisms are available to (b) define and (c) modify these privileges? (d) Who can invoke these mechanisms? (e) Providea list of subjects within the TCB boundary and (f) the list of privileges for each of them.

6. When a subject is created, where do its (a) security attributesand (b) privileges originate, i.e., how are the security attributesand privileges inherited?

7. List the subjects, if any, which are not controlled by the TCB.

2.2 OBJECTS

An object is a passive entity that contains or receives information.Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes.

C1:

1. Provide a list of objects within the TCB (e.g., authentication database, print queues).

2. List the objects in your system that are protected by the Discretionary Access Control (DAC) mechanisms.

3. (a) List the objects that are not protected by the DAC mechanism. (b) Why are they not protected? (c) Describe othermechanisms used to isolate and protect objects.

4. (a) List other resources which are not protected by the DAC mechanism. (Examples include temporary data files accessible only to the file's owner). (b) Why are they not protected byDAC? (c) Describe the mechanisms that are used to isolate and protect these resources.

2.4 SOFTWARE

The TCB software consists of the elements that are involved in enforcing the system security policy. Examples of TCB elements include: kernel, interrupt handlers, process manager, I/O handlers, I/O manager, user/process interface, hardware and command languages/interfaces (for system generation, operator, administrator, users, etc.). The security kernel is the hardware, firmware and software elements of the TCB that are involved in implementing the reference monitor concept, i.e., the ones that mediate all access to objects by subjects.

C1:

6. List all the privileges a process can have. Include the privileges based on the process or user profile, process or user name, or process or user identification.

7. How are a process's privileges determined?

2.5 DISCRETIONARY ACCESS CONTROL (DAC) POLICY

C1:

1. What mechanisms are used to provide discretionary access controls? (Examples of mechanisms are: access control lists, protection bits, capabilities, etc.)

2. (a) Can the access be granted to the users on an individual user basis? (b) If so, how?

3. (a) How is a group defined? (b) What mechanisms are used to administer groups (i.e., to create or delete groups or to add or delete individual users from a group)? (c) Who can invoke these mechanisms? (d) What privileges are necessary to invoke these mechanisms?

4. How can the access be revoked on an individual user basis?

5. How can the access be revoked on a group basis?

6. List any objects that can be accessed by users excluded from the DAC policy (e.g., IPC files, process signaling/synchronization flags)?

7. For each TCB object identified in question 1, section 2.2, describe the DAC mechanism which protects that object.

8. (a) List the access modes supported by the system (e.g., read, write, delete, owner, execute, append). (b) Briefly describethe meaning of each access mode for each object.

9. (a) Are conflicts between user and group access detected?(b) If so, how are the conflicts resolved?

10. For each object, list when changes in DAC permissions become effective.

C2:

11. (a) Can the access be granted to groups of individuals? (b)If so, how?

12. (a) What are the initial access permissions when an object is created? (b) Can the initial access permission be changed? If so, (c) by whom (e.g., user owner, system administrator, others)and (d) how?

13. (a) Can different initial access permissions be specified for different users, or is this a system-wide setting? If the former,(b) by whom and (c) how?

14. (a) Who can grant the access permissions to an object after the object is created? (Examples include creator, current owner, system administrator, etc.) (b) How is the permission granted?

15. (a) Can the ability to grant permissions be passed to anotheruser? If so, (b) by whom and (c) how? Under what circumstances canthe previous owner of the privilege retain it?

B3:

16. (a) Can access be denied to the users on an individual user basis, i.e., exclude individual users? (b) If so, how?

17. (a) Can the access be denied to groups of individuals? (b) Ifso, how?

2.13 OTHER ASSURANCES

C1:

5. (a) List separately the functions that can be performed by eachof the trusted users (e.g., operator, security administrator, accounts administrator, auditor, systems' programmer). (b) For eachof these persons/roles, list the system data bases that can be accessed and their access modes. (c) Also list the privileges provided to each of these roles.

6. (a) How does the TCB recognize that a user has assumed one of the above-mentioned trusted roles? (b) Which of the above-mentioned functions can be performed without the TCB recognizing this role?

Required Readings

TCSEC85 National Computer Security Center, *Department of Defense Trusted Computer Security Evaluation Criteria*, DoD 5200.28-STD, December 1985.

Sections 2.1.1.1, 2.2.1.1, 3.1.1.1, 3.2.1.1, 3.3.1.1, and 4.1.1.1 contain the DAC requirements, which are summarized on pages 99-100. Section 5.3.1.2 describes the control objectives of the discretionary security policy. Section 7.3.3 describes the relationship between policy and the criteria for discretionary security.

INTERP94 National Computer Security Center, *The Interpreted TCSEC Requirements*, (quarterly).

The following Interpretations are relevant to DAC:

I-0002	Delayed revocation of DAC access
I-0020	DAC authority for assignment
I-0222	Passwords not acceptable for DAC
I-0312	Set-ID and the DAC requirement
C1-CI-06-84	Discretionary Access Control
C1-CI-03-85	Discretionary Access Control
C1-CI-01-86	Discretionary Access Control
C1-CI-03-86	DAC by Default
C1-CI-03-89	DAC Public Objects

Gasser88 Gasser, M., *Building a Secure Computer System*, Van Nostrand Reinhold Co., N.Y., 1988.

Sections 3.5 and 3.5.1 introduce DAC. Section 5.4 gives a fair description of user roles. All of Section 6.2 (6.2.1-6.2.5) shouldbe read for a description of DAC. Section 11.3 talks about protected subsystems and 11.6 talks about capability systems.

DAC87 National Computer Security Center, *A Guide to Understanding Discretionary Access Control in Trusted Systems*, NCSC-TG-003, Version 1, 30 September 1987.

This tutorial is fairly broad and describes many areas and aspects of DAC. It should be read in its entirety.

Saltzer75 Saltzer, J.H. and Schroeder, M.D., "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, Vol. 63, No. 9, pp. 1278-1308, September 1975.

This paper provides a detailed description of a number of DAC mechanisms and the issues that pertain to them. Although the paper is somewhat dated, it should be read in its entirety.

Supplemental Readings

None.

Other Readings

Bell76 Bell, D.E. and La Padula, L.J., *Secure Computer Systems: Unified Exposition and Multics Interpretation*, MTR-2997, Rev. 1, MITRE Corporation, Bedford, MA, March 1976.

Downs85 Downs, D. D., Rub, J. R., Kung, K. C., and Jordan, C. S., "Issues in Discretionary Access Control," *Proceedings of the 1985 IEEE Symposium on Security and Privacy*, pp. 208-218, April 1985.

This paper discusses the types of mechanisms that can be used to implement DAC. It also covers the access types that can be controlled by a DAC mechanism.

Gligor87 Gligor, V., Huskamp, J., Welke, S., Linn, C., and Mayfield, W., *Traditional Capability-Based Systems: An Analysis of Their Ability to Meet The TCSEC*, Institute for Defense Analysis, P-1935, February 1987.

Section 2.2 illustrates the notions of non-circumventability and isolation of a reference monitor built on a capability system. Sections 3.1.4 and 3.2.2 illustrate the difficulties of such systems supporting the TCSEC and suggest solutions.

Gong75 Gong, L., "A Secure Identity-Based Capability System," *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pp. 56-65, May 1989.

While describing a capability system, the author also covers the concepts of "propagation of access rights" and "revocation."

Graham72 Graham, G.S. and Denning, P.J., "Protection -- Principles and Practice," *Proceedings of the AFIPS 1972 SJCC*, 1972.

This paper provides an example of a DAC model and an informal presentation of the model's possible interpretations.

Karger87 Karger, P.A., "Limiting the Damage Potential of Discretionary Trojan Horses," *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, pp. 32-37, April 1987.

This paper describes some inherent vulnerabilities of DAC (specifically Trojan horses) and describes a mechanism that can partially relieve this problem.

Karger89 Karger, P.A., "New Methods for Immediate Revocation," *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pp. 48-55, May 1989.

This paper describes two new immediate revocation methods.

Sandhu86 Sandhu, R.S. and Share, M.E., "Some Owner Based Schemes with Dynamic Groups in the Schematic Protection Model," *Proceedings of the 1986 IEEE Symposium on Security and Privacy*, pp. 61-70, April 1986.

This paper investigates a variety of policies for defining group membership dynamically in the context of a simplified file system. It specifies the policies using the Schematic Protection Model (SPM). The investigation reveals that there are many policy options and demonstrates how SPM is used to precisely specify these options.

Vinter88 Vinter, S.T., "Extended Discretionary Access Controls," *Proceedings of the 1988 IEEE Symposium on Security and Privacy*, pp. 39-49, April 1988.

This paper presents a DAC mechanism proposed for SDOS. SDOS is an object-oriented system that uses ACLs to grant access to objects.

Wisema86 Wiseman, S., "A Secure Capability Computer System," *Proceedings of the 1986 IEEE Symposium on Security and Privacy*, pp. 86-94, April 1986.

A secure computer system based on a capability architecture is described. Abstract types are used to provide separation and the reference monitor function. By providing a trusted path from the user to security critical operations, full DAC and MAC is enforced.